RBCorr: Response Bias Correction in Language Models

Om B. Bhatt
Cognitive Sciences
University of California, Irvine
om.bhatt@uci.edu

Anna A. Ivanova
Psychological and Brain Sciences
Georgia Institute of Technology
a.ivanova@gatech.edu

Abstract

Language models (LMs) are known to be prone to response biases, which present as option preference biases in fixed-response questions. It is therefore imperative to develop low-cost and effective response bias correction methods to improve LM performance and enable more accurate evaluations of model abilities. Here, we propose a simple response bias correction strategy, `RBCorr`, and test it on 12 open-weight language models using yes-no, entailment, and multiple choice questions. We show that response bias is prevalent in LMs pre-correction and that `RBCorr` effectively eliminates bias and boosts model performance. We also explore the generalizability of bias behavior across models, datasets, and prompt formats, showing that LogProbs-based correction is highly dependent on all three of these aspects. Overall, `RBCorr` is an easy-to-use method that can boost the performance of smaller LMs and ensure that LM performance on closed-response benchmarks aligns more closely with their true capabilities. Code and datasets: https://github.com/ombbhatt/rbcorr_bias_correction.git

1 Introduction

In fixed-response evaluation settings, language models (LMs) have been reported to suffer from response bias (e.g., Pezeshkpour and Hruschka, 2023; Tjuatja et al., 2024; Zhao et al., 2021), which reduces their performance and viability as real-world reasoners. Recent mitigation efforts have looked towards LogProbs manipulations as a general strategy for its ease of accessibility and low cost, but we believe that further advancement also requires a systematic exploration of the experimental setup (i.e. the model, dataset, and prompt scheme being used), as well as tracking bias separately from accuracy to gain more nuanced insights on the effects of applying correction, and finally, new manipulation strategies themselves to try and improve upon existing methods.

This paper investigates and corrects response biases in open-weight LMs for fixed-response questions. We test LMs across a variety of reasoning tasks and prompt formats, track both bias and accuracy, and propose a simple LogProbs-based correction method that can be applied to multiple question types, effectively mitigating bias and increasing or maintaining accuracy. Additionally, we compare our method to existing correction methods, characterize models' baseline label preferences, and perform further analysis to reveal that LogProbs values are highly context-specific to the model, dataset, and prompt formats being used. Our results show that a straightforward calibration-based bias correction strategy, akin to approaches in traditional machine learning (e.g., Saerens et al., 2002; Zadrozny and Elkan, 2002), can be effectively applied to modern LMs to improve their performance and enable more faithful evaluations of their capabilities.

2 Related Work

Several bias evaluation papers provide behavior characterizations across models and datasets. Pezeshkpour and Hruschka (2023) demonstrate that the order in which options are presented can drastically impact model performance in ICL settings. Salecha et al. (2024) demonstrates that LMs shift their responses to be more socially desirable when they are provided with enough questions to self-infer that they are being socially evaluated. Tjuatja et al. (2024) develop a human-survey like dataset to test various LMs on five types of biases and show that models in general fail to reflect known human-like response patterns, highlighting the risk of using them as human proxy reasoners.

Our work falls more specifically into a smaller family of recent works that aim to mitigate bias using theoretically-motivated LogProbs manipulation methods. Two of these methods — Contextual Calibration (Zhao et al., 2021) and Batch Calibra-

Proceedings of the Fifth Workshop on Generation, Evaluation and Metrics (GEM), pages 540–553

tion (Zhou et al., 2024) — are used for comparison with our method, but there are other proposed methods as well, such as Domain Calibration (DC) (Fei et al., 2023) and Prototypical Calibration (PC) (Han et al., 2022).

We decide to exclude comparison to them because the (Zhou et al., 2024) work shows that BC performs better in general compared to PC and DC, making BC sufficient for performance comparison. We perform one additional auxiliary comparison between our method and the PriDe method introduced by (Zheng et al., 2024) — this method is not similar in principle to the other methods in this family, since it scales overhead compute cost with correction efficacy, but still aims to correct label bias using prior estimation. PriDe comparison results are provided in Appendix C.

3 Approach

Our approach is based on extracting token log probabilities (LogProbs) from the last layer of an LM, prior to softmax transformation and final token sampling. In a bias-free model, the average LogProbs for different response options should be equal (as long as the dataset is class-balanced). Our goal is to measure the deviation between this uniform distribution and empirically observed model LogProbs scores (i.e., response bias) and to correct it by applying a correction term to an LM's LogProbs values prior to response generation.

3.1 Extracting Model Response

For each type of question, we define single-token response formats in the prompts. We then extract the log-probability values for the appropriate set of response tokens for each dataset item depending on the question type (we test on 2-choice, 3-choice, and 4-choice questions). The token with the highest log-probability value is recorded as the model's response. To account for response variation, we consider the whitespace-prepended version of each token as well (we log-sum-exp the LogProbs values for '␣Yes' and 'Yes' when evaluating 2-choice questions, for example.)

3.2 Measuring Response Bias

We use the two metrics suggested by Reif and Schwartz (2024) to measure model bias: **Total Variation Distance (TVD)** and **Relative Standard Deviation (RSD)**. For both metrics, one value is calculated between the ground-truth label distribution and the model response distribution, both before and after applying correction. For both metrics, a value of 0 indicates identical distribution to the ground-truth, i.e. perfectly unbiased.

For a ground-truth dataset label distribution G and model response label distribution M, TVD is calculated as:

$$\textbf{TVD}(G, M) = \frac{1}{2} \sum_{x \in X} |G(x) - M(x)| \quad (1)$$

RSD is defined as the standard deviation of the model's class-wise accuracy divided by its mean accuracy acc on the entire evaluation data (Reif and Schwartz, 2024):

$$\textbf{RSD} = \frac{\sqrt{\frac{1}{|X|} \sum_{i=1}^{|X|} (acc_i - acc)^2}}{acc} \quad (2)$$

Where X denotes the relevant option label space.

The motivation behind reporting both measures is to capture explanative power on the nature of the bias correction. A model's bias could be revealed either by observing biased output probabilities *on average* (TVD-sensitive) or by observing biased *per-class* output probabilities (RSD-sensitive) (Reif and Schwartz, 2024). Intuitively, RSD is high when the disparity in performance among classes is high, while TVD is high (and thus flags cases) when class-wise performance is balanced but output probabilities are biased on average. We use TVD as the primary metric in our plots because it is more sensitive to overall distribution shifts.

For simplicity of the TVD metric calculation, we pick our test datasets to be class-balanced, so that the ground-truth is the uniform distribution, and we can directly use the entire model distribution $M(x)$ in (1). **In real-world application** with potentially non-balanced datasets, (Reif and Schwartz, 2024) suggest sampling a balanced subset of in-distribution items from the test set to use in place of $M(x)$ for the equivalent calculation.

3.3 Applying RBCorr Correction

Our correction method, `RBCorr`, applies a mean-normalization to individual item LogProbs values by using a small held-out class-balanced calibration set for mean LogProbs estimation for each response option. To perform this correction, we first extract model responses and the options' LogProbs values for each item in a dataset. We then sample a small

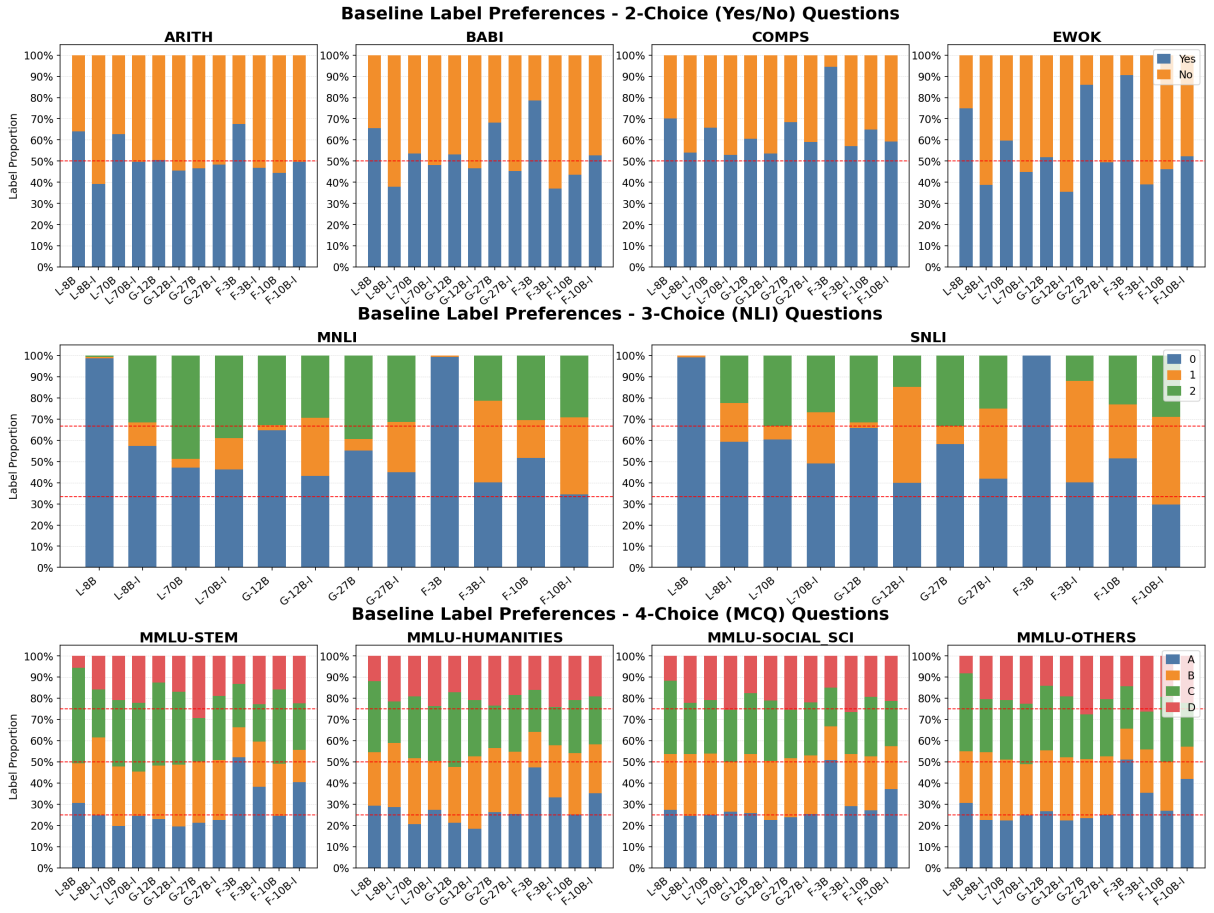

Figure 1: Baseline model response label distribution for all models on all datasets using the Instruction-only prompt format. Red dotted lines indicate ground-truth uniform distribution intervals.

calibration set of questions from the entire dataset, and calculate the mean of the LogProbs value for each of the option tokens in that set. Importantly, we enforce class-balance when sampling the calibration set, to ensure that the means estimation is not skewed by over- or under-representation of any class of questions. We finally subtract these means from the corresponding tokens' LogProbs for every item in the evaluation set, i.e., all questions *outside* of the calibration set. We extract model responses from this corrected set of LogProbs to record our 'correction-applied' results. This correction requires no overhead computation to perform since it only requires the baseline LogProbs values for adjustment.

3.4 How Many Questions to Calibrate On?

To explore this, we perform our correction method using a range of fixed calibration set sizes: $\boldsymbol{N} = \{\mathbf{24, 60, 120, 180, 240}\}$ questions. To achieve complete evaluation coverage, we apply RBCorr in a k-fold fashion — we sweep through the dataset k times using separate balanced samplings of size $n \in N$ items as the calibration set used to correct all other items. Since we want to compare performance across different calibration set sizes, we fix $\boldsymbol{k = 5}$ to put an upper bound on the number of independent correction estimates calculated for the items. We average these corrected values to get our final corrected value per item. We plot the accuracy achieved using these different calibration set sizes in Fig. 3.

In our calibration size variation results, we observe an immediate and sustained increase in accuracy across all datasets beginning from the smallest set size. Tracking the average corrected accuracy across all 12 models at each size shows that `RBCorr` is functionally **unaffected by calibration set size.** The fact that correction efficacy stays consistent across such a large range of calibration sizes indicates that response bias within a model-dataset-prompt configuration is possibly low-variance by nature, causing our estimator to converge quickly. We fix `RBCorr` at $\boldsymbol{n = 60}$ in our main experiments.

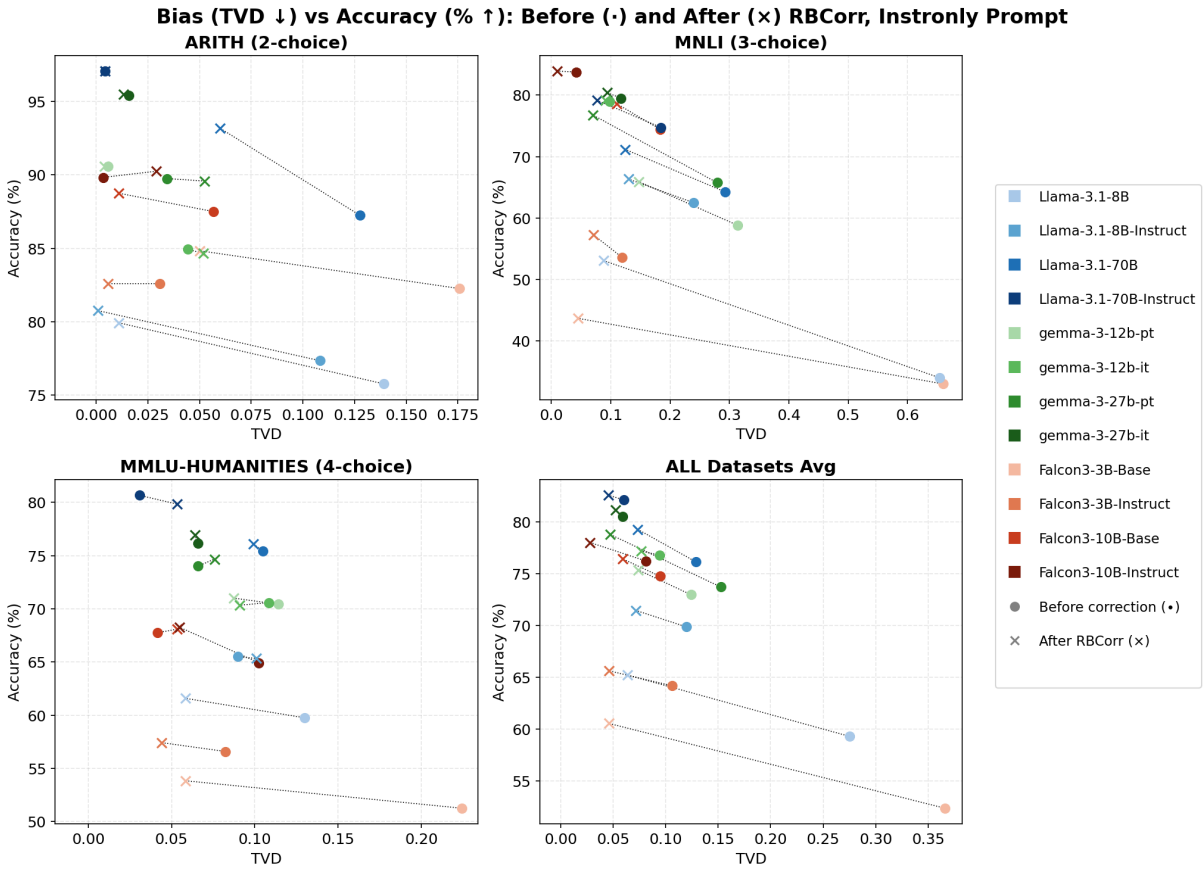

Figure 2: Scatterplots showing per-model bias (TVD; ↓ is better) and accuracy (%; ↑ is better) before and after applying `RBCorr` correction. We show results on one dataset per each question-type; the bottom-right plot shows results averaged across all 10 datasets.

4 Experimental Setup

Here we describe the datasets, models, and prompt formats used to test our correction method and perform subsequent experiments for further analysis.

4.1 Datasets

We test LMs on datasets with two, three and four answer options, covering a wide variety of knowledge domains. We release these datasets alongside our code.

2-Choice (`'Yes'/'No'`): We use subsets of four existing datasets converted into Yes-No format: **(1) ARITH**, an addition and subtraction problem set from the BIGBench dataset (bench authors, 2023); **(2) BABI**, a 12-domain reading comprehension task set (Weston et al., 2015) (e.g., `"Julie travelled to the park. Is Julie in the {bedroom / park}?"`); **(3) COMPS**, a question set testing basic property inheritance in minimal pairs (e.g., `"Does {an iguana / a trolley} bask in the sun?"`) (Misra et al., 2023); and **(4) EWOK**, an 11-domain question set testing contextual world knowledge (e.g., `"Chao is making Yan's job {easier / harder}. Is Chao {helping / hindering} Yan?"`) (Ivanova et al., 2025).

3-Choice (`'0'/'1'/'2'`): We use the **(5) MNLI** (Williams et al., 2018) and **(6) SNLI** (Bowman et al., 2015) datasets. Both datasets consist of Recognizing Textual Entailment (RTE) questions, where a premise and a hypothesis sentence are provided, and the task is to classify whether the hypothesis entails, is neutral to, or contradicts the premise (e.g., `"Premise: A woman is reading a book in the library. Hypothesis: A woman is swimming."` is a contradiction, i.e., `'2'`.)

4-Choice (`'A'/'B'/'C'/'D'`): We sample from the **MMLU** (Hendrycks et al., 2021) collection of datasets. MMLU consists of 57 datasets spanning various topics, which can be broadly grouped into four large datasets: **(7) Humanities**, **(8) Social Sciences**, **(9) STEM**, and **(10) Others**.

Dataset Construction: To maintain experimental consistency, we pick **1200** class-balanced item

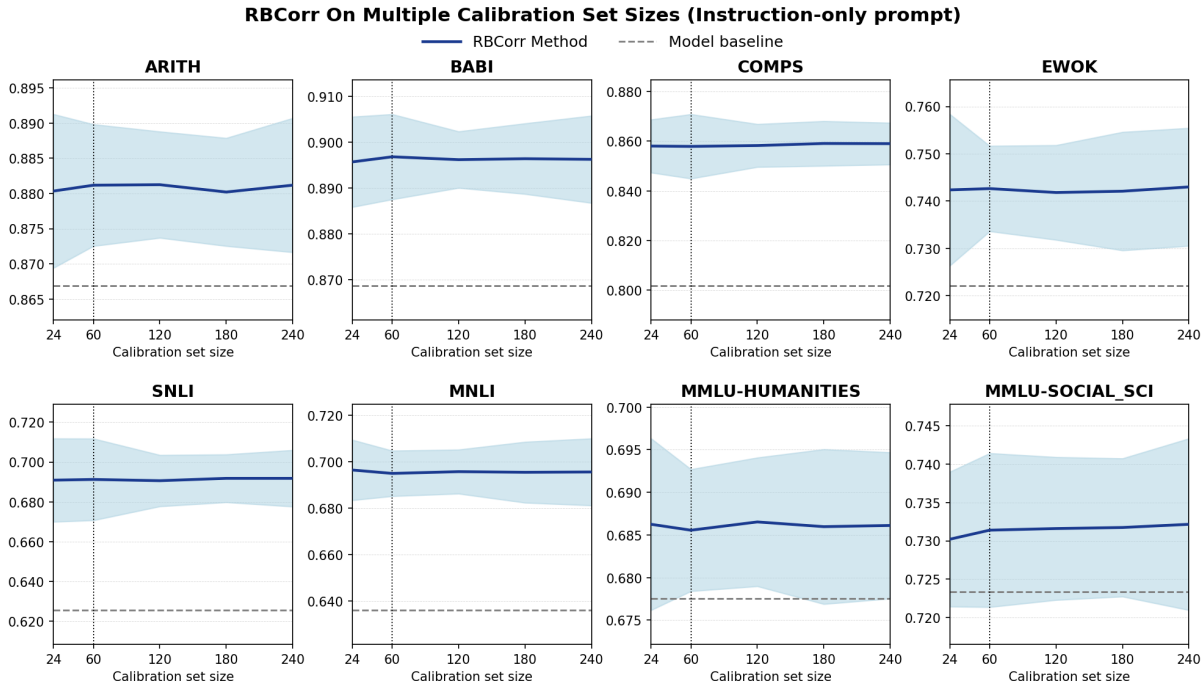

Figure 3: Lineplot showing the corrected accuracy (solid blue line) when using differently-sized calibration sets and $k = 5$ folds for RBCorr, averaged across all 12 LMs using the Instruction-only prompt. We find that RBCorr's effect on accuracy is **insensitive to calibration set size**. Light blue shading shows the 2σ ($\approx 95\%$ CI) band covering the mean of the accuracies of the individual k-fold correction passes done per model-dataset configuration.

sets for each of the 10 datasets, totaling 4800, 2400, and 4800 test questions for 2-,3-, and 4-choice types respectively. In datasets with multiple subdomains, we maintain as equal subdomain representation as possible. The class-balance allows us to (A) easily visualize baseline label preferences as deviations from the uniform distribution (as in Fig. 1), and (B) maintain a straightforward TVD calculation (as explained in 3.2). However, this is only done for analytical simplicity; it possible to do equivalent analysis and correction without class-balance as well.

4.2 Models

We test 12 LMs across 3 model families: Falcon3, Gemma3, and Llama3.1. Each family contains four LMs — two pairs of smaller and larger models (e.g. 3B/10B for Falcon3 or 8B/70B for Llama3.1). Each such pair represents a base and an instruction-tuned version of an LM. For instance, the two pairs that comprise the Gemma3 family are {(Gemma3-12B, Gemma3-12B-IT), (Gemma3-27B, Gemma3-27B-IT)}. This model test set construction allows us to observe how bias may change with model size and instruction tuning while keeping the model architecture constant.

4.3 Prompt Complexity

We record model responses across three prompt formats varying in level of complexity:

1. **'Zeroshot'**: Only test question is presented,
2. **'Instruction-only'**: One-line task instruction precedes the test question,
3. **'Fewshot'**: One-line task instruction and two examples precede the test question.

See Appendix A for the full prompts.

5 Experiment 1: Models Are Biased

Figure 1 shows the label proportions for all models tested on all question-types, grouped by dataset, using the fewshot prompt format. The figure reveals that baseline model results (i.e. without applying correction) show varying degrees of inherent label bias. We see all models bias towards 'Yes' on COMPS and show mixed biases on other yes-no datasets. Interestingly, nearly all non-instruct models are 'Yes'-biased, and instruction-tuning results in reduced or flipped bias behavior. In NLI datasets, we see a general strong bias towards option '0', with the Llama3.1-8B and Falcon3-3B models purely responding with '0', while their in-

Metric	Method	Zeroshot	Instr-only	Fewshot
		$\overline{\Delta}$ Baseline	$\overline{\Delta}$ Baseline	$\overline{\Delta}$ Baseline
Accuracy (%) (↑ better)	CC	-5.53^{***}	-0.27 n.s.	-0.33 n.s.
	BC	$+2.45^{***}$	$+2.58^{***}$	$+1.37^{***}$
	RBCorr	$\mathbf{+2.66}^{***}$	$\mathbf{+2.73}^{***}$	$\mathbf{+1.46}^{***}$
TVD (↓ better)	CC	$+0.0843^{***}$	$+0.0141$ n.s.	$+0.0193$ n.s.
	BC	$\mathbf{-0.1176}^{***}$	$\mathbf{-0.0817}^{***}$	$\mathbf{-0.0367}^{***}$
	RBCorr	-0.1170^{***}	-0.0815^{***}	-0.0347^{***}
RSD (↓ better)	CC	$+0.1734^{***}$	$+0.0187$ n.s.	$+0.0361^{*}$
	BC	$\mathbf{-0.1794}^{***}$	-0.1209^{***}	$\mathbf{-0.0417}^{***}$
	RBCorr	-0.1792^{***}	$\mathbf{-0.1217}^{***}$	-0.0393^{***}

Table 1: Comparison all three correction methods' effects relative to 'baseline' (pre-correction) performance metrics. Values are averages across all model and dataset combinations for a specific prompt format. *RBCorr yields the biggest accuracy gains*. Here, *** p<0.05, ** p<0.01, *** p<0.001, n.s. non-significant** (two-tailed).

struct counterparts introduce other responses. In 4-choice datasets, we see relatively more uniform option preference across all models.

Across all models, we see that the bigger-size versions of models usually yield more uniform label distributions relative to their smaller counterpart. We see the same trend with the instruction-tuned version compared to the base version of the model. *This supports the claim that using both bigger and instruction-tuned versions of models can inherently help reduce response bias.*

6 Experiment 2: `RBCorr` Improves Performance

Figure 2 shows the accuracy and bias changes for all models after applying our correction (we plot the mean corrected accuracy and bias across 5 correction passes using calibration sets of size 60, using the Instructions-only prompt format). We pick one representative dataset from each question type for three plots, and the fourth plot shows the average changes across all ten datasets.

We consistently see a reduction in bias value with either a higher or maintained accuracy across all models on all datasets. Models show the biggest accuracy increase in the 3-choice MNLI dataset and smallest in 4-choice MMLU-STEM. The final average scatterplot shows that the smaller-sized model pairs from the Falcon3 and Llama3.1 families had the biggest overall accuracy gains from applying the correction. We see several cases of $10-15\%$ accuracy gains in the 2-choice and 3-choice datasets (see Appendix D). *This indicates that our bias correction can significantly recover task performance lost due to response bias, and that it is most effective for smaller-sized models.*

Significance of bias correction beyond accuracy improvement: While some models in the individual-dataset plots show low accuracy gains, we believe that the correction method is still worth applying for model evaluation purposes, because it allows for a fairer comparison of reasoning capabilities after removing the model's inherent label response bias. Removing the label bias and observing change in accuracy gives us important diagnostic information about label bias being a potential factor affecting the model's output performance. If model performance is in fact influenced by label bias, then removing the bias may uncover latent performance that was hidden as a result of the label priors. Conversely, if label bias is not a substantial driving factor for model performance, then removing label bias will not do much for improving accuracy, but it does still remove label bias as a potential confound to performance, allowing an evaluator to test for other biases or non-bias related problems affecting model performance.

7 Experiment 3: Comparing `RBCorr` With Existing Methods

Here we describe two existing LogProbs-based bias mitigation methods and perform them in our experimental setup for comparison with `RBCorr`.

(1) Contextual Calibration (CC) (Zhao et al., 2021): This correction involves collecting and averaging model output probabilities using a

few "content-free" input prompts (using "N/A", "[MASK]", and the empty string), and then using them to adjust test item probabilities via an affine matrix operation; this is equivalent to dividing the test item output probability with the mean content-free output probability.

(2) Batch Calibration (BC) (Zhou et al., 2024): Like RBCorr, BC calculates mean LogProbs for each response option using within-dataset samples to estimate the correction term. However, our method differs from BC in two aspects. First, BC does not enforce class-balance in its random sampling of the batch items, which introduces the possibility of skewed averages. Second, BC does not maintain any train-test separability: the items used to calculate the correction terms are the same items that the correction term is applied to, which is detrimental to evaluation integrity and reproducibility in both open models as seen here and more generally via 'indirect' data leakage in closed-source models (Balloccu et al., 2024). In RBCorr, the calibration and evaluation items are strictly separate. We use a batch size of 60 for BC, matching our RBCorr calibration set size, and independently calculate the correction term for each batch.

Table 1 shows the effects of all three bias correction methods on accuracy, TVD and RSD for different prompt formats. We observe that RBCorr gets the highest accuracy gain (with the highest gain achieved on the Instructions-only prompt), while BC gets highest TVD/RSD reduction relative to baseline. This implies that BC potentially overcorrects, but that the extra correction does not translate to accuracy. The class balance in RBCorr's term likely supports the model's weakest-performing class more selectively, thus increasing performance without affecting the response distributions for other classes as much (i.e. a slightly lower TVD/RSD change).

Additionally, we do a pairwise comparison of performance affects for all three correction methods on the Instr-only prompt:

Method	Acc. (%) (↑)	TVD (↓)	RSD (↓)
RBCorr vs CC	$+3.00^{***}$	-0.0956^{***}	-0.1404^{**}
RBCorr vs BC	$\mathbf{+0.15}^{**}$	$+0.0003$ **n.s.**	-0.0008 **n.s.**
BC vs CC	$+2.85^{***}$	-0.0959^{***}	-0.1396^{***}

We observe that RBCorr's edge in accuracy gain over BC is statistically significant, while BC's edge in bias reduction over RBCorr (as observed here via the TVD score only) is insignificant. The statistical significance of accuracy gain compared to BC holds for the other two prompt settings as well.

Notably, our implementation of CC worsened (increased) bias according to both measures across all prompt types, and negatively affected accuracy in the zeroshot case while having no significant accuracy effect in the other two cases. One potential reason might simply be that the model's representations of context-free strings are inherently separate from what we may intuit as a neutral point to extract non-biased response token values, and thus are poor correction value candidates.

Across all metrics, fewshot results have the smallest changes from baseline, suggesting that in-context examples results in a relatively less-biased baseline starting point compared to the other prompts that do not contain examples.

See Appendix D for the full per-model, per-dataset results of applying each correction method on the instruction-only prompt.

8 Experiment 4: Correction Is Not Transferrable

A unique aspect of RBCorr is that we calculate a single static set of mean LogProbs values for answer options in a dataset, which serve as 'correction terms.' Once LogProbs values for any configuration (model, dataset, prompt format) are generated, the resulting bias correction terms can be applied to other LogProbs sets. This enables testing the **transferability** of correction terms across **models**, **datasets**, and **prompts**, effectively examining bias consistency across these modalities.

We test transfer efficacy by transferring correction terms across each modality independently (e.g., a cross-model correction will only source the term from other same-prompt, same-dataset runs). We conduct $k = 5$ correction runs with random calibration set sampling at a fixed size of 240 questions. Evidence of reliable correction in these transfer experiments should tell us whether response bias is most stably captured by the model, dataset, or prompt being used for evaluation.

Table 2 shows that correction terms fail to transfer successfully across varying configurations, despite changing only one modality at a time and enforcing systematic constraints (only within-family

Modality	Total Pairs	Succ. Transfers	Avg ΔAcc (success pairs)	Avg ΔTVD (success pairs)
Cross-Dataset	912	262 (28.73%)	0.0316	0.1011
Cross-Model	1008	176 (17.46%)	0.0119	0.0413
Cross-Prompt	624	90 (14.42%)	0.0113	0.0454

Table 2: Quantifying successful transfer of RBCorr's bias correction term across all three modalities for all valid source-target configuration pairs. Successful transfer correction is defined as ≥80% preservation of bias reduction and accuracy increase compared to non-transfer correction on the target configuration.

transfer for cross-model, and only within-question-type transfer for cross-dataset). We define a "successful" transfer correction as one that achieves either (1) at least ≥80% of the model performance improvements or (2) at most 80% of the model performance damages that were yielded on applying the 'regular' (i.e. non-transfer) RBCorr on the tested target configuration, and depending on whether that 'regular' correction originally resulted in an improvement or a harm to the model performance metrics.

Based on this success criteria, all three transfer modalities perform poorly. Notably, Cross-dataset transfer shows considerably higher success (28.73%) relative to the other two, indicating that bias behavior can be carried most stably across datasets by virtue of the behavior-capturing power of the model and prompt information. Conversely, the dataset is the weakest anchor for capturing bias, i.e., it would be hard to find any response behavior generalities between different models and prompts by looking at their results on the same dataset. In the bigger picture, however, the fairly poor success rates across all modalities demonstrate that *in-context calibration is a requirement for reliable bias correction using* RBCorr. Unlike Zheng et al. (2024), we find limited transfer efficiency; caution is therefore warranted when trying to derive a universal correction term.

As an auxiliary analysis, Appendix B presents transfer results for three individual configurations. While not supporting general inference, it reveals transfer asymmetry: transferring from configuration A → B yields different effects than B → A.

9 Conclusion & Future Work

In this paper, we quantify LLM response bias in three kinds of closed-response questions using a experimental setup consisting of various datasets, models, and prompt schemes. We then propose a LogProbs-based bias correction method, RBCorr, that effectively reduces measured bias and improves accuracy, and compare it to similar existing methods to demonstrate its efficacy. Finally, we conduct an analysis to show that LogProbs values are highly specific to model, dataset, and prompt configurations, and bias estimations cannot reliably generalize over any of the three settings.

RBCorr provides value both as a performance improver and an evaluation diagnostic technique. For performance improvement, a system can store a set of pre-computed calibration value for a given prompt and question type and apply them dynamically in online inference settings. The performance improvement we observe is particularly relevant to small and medium-sized LMs, offering an opportunity to use these lightweight cheap models for large-scale but relatively straightforward tasks (e.g., closed-form text labeling). For evaluation, our method helps uncover latent model performance and determine whether response bias hampers its performance on a task it could otherwise solve. In the age of skepticism of LM evaluations (Banerjee et al., 2024; Cao et al., 2025), we posit that benchmark-based evaluations might still have value, as long as they are designed to measure general capabilities rather than specific task performance, maintain genuine train/test set separability (Balloccu et al., 2024), and are coupled with debiasing techniques that help uncover latent model performance.

Future work could explore using corrected probabilities as a tuning set to inherently debias the model's outputs, i.e., training a model to adjust its label token probabilities to align with previously-extracted and corrected probabilities, leading to better performance without having to apply post-hoc correction to the LogProbs results. We also point toward mechanistic interpretability approaches for measuring bias across model layers (e.g., Gupta

et al., 2025), tracing the origin of the bias in model weights, and correcting the bias at its source rather than at the last layer. Finally, we consider response bias to be a valuable test case for bias exploration and correction, which can then be extended to open-form responses and reasoning traces. Finally, we also hope to encourage further work leveraging LogProbs for model analysis and potential improvements as a way to promote open access to newer LMs.

Limitations

Our method requires at least partial access to class labels to be able to perform balanced calibration set sampling. However, we demonstrate that even a small-sized calibration set achieves performance improvement. In the case of complete lack of labels, one way to apply `RBCorr` would be to generate labels for a held-out test set of items using a SOTA LLM and using them as proxy ground-truth labels. Our method is only applicable to open-source models, for which LogProbs values of tokens are accessible. For closed models, one may try to estimate the bias by sampling responses with high temperature, but that method may be less precise. Additionally, we test transformer-only models up to 70B parameters in size; we do not report on the response behavior and correction efficacy on larger models or models based on other architectures, although we expect similar trends to hold.

Acknowledgments

We thank members of the LIT lab who provided feedback on earlier stages of this work. We also thank Han Zhou (Zhou et al., 2024) for clarifying high-level details for implementing the Batch Calibration correction in our experiments. This work was supported in part through the Partnership for an Advanced Computing Environment (PACE) at the Georgia Institute of Technology, RRID:SCR_027619.

References

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondrej Dusek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 67–93, St. Julian's, Malta. Association for Computational Linguistics.

Sourav Banerjee, Ayushi Agarwal, and Eishkaran Singh. 2024. The vulnerability of language model benchmarks: Do they accurately reflect true LLM performance? *arXiv preprint arXiv:2412.03597*.

BIG bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. *Preprint*, arXiv:1508.05326.

Yixin Cao, Shibo Hong, Xinze Li, Jiahao Ying, Yubo Ma, Haiyuan Liang, Yantao Liu, Zijun Yao, Xiaozhi Wang, Dan Huang, and 1 others. 2025. Toward generalizable evaluation in the LLM era: A survey beyond benchmarks. *arXiv preprint arXiv:2504.18838*.

Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. 2023. Mitigating label biases for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14014–14031, Toronto, Canada. Association for Computational Linguistics.

Akshat Gupta, Jay Yeung, Gopala Anumanchipalli, and Anna Ivanova. 2025. How do LLMs use their depth? *arXiv preprint arXiv:2510.18871*.

Zhixiong Han, Yaru Hao, Li Dong, Yutao Sun, and Furu Wei. 2022. Prototypical calibration for few-shot learning of language models. *Preprint*, arXiv:2205.10183.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Anna A. Ivanova, Aalok Sathe, Benjamin Lipkin, Unnathi U. Kumar, Setayesh Radkani, Thomas H. Clark, Carina Kauf, Jennifer Hu, R. T. Pramod, Gabriel Grand, Vivian C. Paulun, Maria Ryskina, Ekin Akyürek, Ethan G. Wilcox, Nafisa Rashid, Leshem Choshen, Roger Levy, Evelina Fedorenko, Joshua Tenenbaum, and Jacob Andreas. 2025. Elements of world knowledge (ewok): A cognition-inspired framework for evaluating basic world knowledge in language models. *Transactions of the Association for Computational Linguistics*, 13:1245–1270.

Kanishka Misra, Julia Taylor Rayz, and Allyson Ettinger. 2023. Comps: Conceptual minimal pair sentences for testing robust property knowledge and its inheritance in pre-trained language models. *Preprint*, arXiv:2210.01963.

Pouya Pezeshkpour and Estevam Hruschka. 2023. Large language models sensitivity to the order of options in multiple-choice questions. *Preprint*, arXiv:2308.11483.

Yuval Reif and Roy Schwartz. 2024. Beyond performance: Quantifying and mitigating label bias in LLMs. In *Proceedings of the 2024 Conference of*

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6784–6798, Mexico City, Mexico. Association for Computational Linguistics.

Marco Saerens, Patrice Latinne, and Christine Decaestecker. 2002. Adjusting the outputs of a classifier to new a priori probabilities: a simple procedure. *Neural computation*, 14(1):21–41.

Aadesh Salecha, Molly E. Ireland, Shashanka Subrahmanya, João Sedoc, Lyle H. Ungar, and Johannes C. Eichstaedt. 2024. Large language models show human-like social desirability biases in survey responses. *Preprint*, arXiv:2405.06058.

Lindia Tjuatja, Valerie Chen, Tongshuang Wu, Ameet Talwalkwar, and Graham Neubig. 2024. Do LLMs exhibit human-like response biases? a case study in survey design. *Transactions of the Association for Computational Linguistics*, 12:1011–1026.

Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards ai-complete question answering: A set of prerequisite toy tasks. *Preprint*, arXiv:1502.05698.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. *Preprint*, arXiv:1704.05426.

Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699.

Tony Z. Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *Preprint*, arXiv:2102.09690.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors. *Preprint*, arXiv:2309.03882.

Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine Heller, and Subhrajit Roy. 2024. Batch calibration: Rethinking calibration for in-context learning and prompt engineering. *Preprint*, arXiv:2309.17249.

A All Input Prompts

Below are all the fewshot prompts used for all datasets. The instruction-only prompt format uses the same prompt scheme, simply without the examples, while the zeroshot prompt format has no text scaffolding and simply feeds in the test item from the dataset to the LM. We directly pass these prompts into the LMs without any evaluation harnesses in order to reduce artifacts from model-specific prompt formatting modifications.

NOTE: Since the 3-choice datasets (SNLI and MNLI) inherently require instructions to define the objective and response format, we only test those datasets using the Instruction-only and Fewshot prompts.

A.1 ARITH

#INSTRUCTIONS
Answer the following yes-no questions:

#EXAMPLE
Question: Is 7 minus 9 equal to 4?
Response: No

#EXAMPLE
Question: Is 17 plus 15 equal to 32?
Response: Yes

#EXAMPLE
Question:

A.2 BABI

#INSTRUCTIONS
Answer the following yes-no questions:

#EXAMPLE
Question: Marshall is in the car. Is Marshall in the building?
Response: No

#EXAMPLE
Question: Nathan is a pianist. Pianists like oranges. Does Nathan like oranges?
Response: Yes

#EXAMPLE
Question:

A.3 COMPS

#INSTRUCTIONS
Answer the following yes-no questions:

#EXAMPLE
Question: Does a blueberry fire bullets?
Response: No

#EXAMPLE
Question: Does a turtle have a hard shell?
Response: Yes

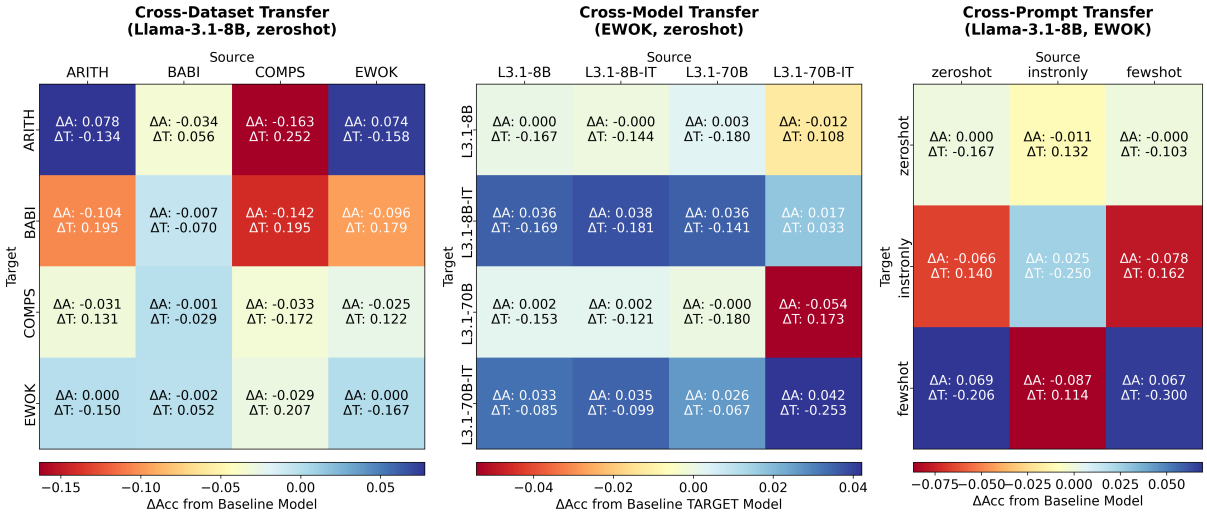

Figure 4: Heatmaps showing transfer correction performance for all three transfer modalities using three specific model-dataset-prompt setups. Gradient indicates accuracy change after applying correction relative to baseline model accuracy.

#EXAMPLE
Question:

A.4 EWOK

#INSTRUCTIONS
Answer the following yes-no questions:

#EXAMPLE
Question: Claire sees something that is fabric. Can Claire pour it?
Response: No

#EXAMPLE
Question: Sally pays salary to Harry. Is Sally Harry's boss?
Response: Yes

#EXAMPLE
Question:

A.5 SNLI and MNLI

#INSTRUCTIONS
Answer the following Recognizing Textual Entailment questions using a single digit. Entailment (0) implies the hypothesis is true given the premise. Neutral (1) implies the premise doesn't provide enough information to determine the hypothesis. Contradiction (2) implies the hypothesis is false given the premise.

#EXAMPLE
Premise: A man is playing a guitar. Hypothesis: A person is making music.
Response: 0

#EXAMPLE
Premise: A woman is reading a book in the library.
Hypothesis: A woman is swimming.
Response: 2

#EXAMPLE:

A.6 MMLU (All Domains)

#INSTRUCTIONS
Answer the following multiple choice questions:

#EXAMPLE
Question: What is the shape of the Earth?
Options: (A) Cone, (B) Cube, (C) Sphere, (D) Cylinder
Response: C

#EXAMPLE
Question: What is the color of the sky?
Options: (A) Red, (B) Blue, (C) Green, (D) Yellow
Response: B

#EXAMPLE
Question:

B RBCorr Transfer Correction – Individual Configuration Heatmaps

We pick one specific configuration to illustrate the effects of `RBCorr` transfer correction in more gran-

ular detail. The configurations for each transfer modality are described below:

1. **Cross-dataset**: Transfer among the 2-choice Yes-No datasets, using Llama3.1-8B as the model and zeroshot as the prompt level,
2. **Cross-model**: Transfer among the Llama3.1 family of models, using EWOK as the dataset and zeroshot as the prompt level,
3. **Cross-prompt**: Transfer among all three prompt levels, using EWOK as the dataset and Llama3.1-8B as the model.

Figure 4 show the results of applying transfer correction in these setups. In general, transfer correction performs poorly and transfer efficacy is asymmetric. These specific transfer instances also allow us to make some model family-level hypotheses; for example, the cross-model results show that both the instruction-tuned Llama3.1 models are consistently good targets for cross-model correction, indicating that bias patterns for them are especially invariant to the source model compared to the dataset or prompt. Such transfer analysis over other individual configuration setups may reveal other potential hypotheses to inform us of unknown model behavior traits.

C Comparing RBCorr with PriDe

Model	ΔRStd% MMLU		ΔAcc.% MMLU	
	Ours 1×	PriDe 1.15×	Ours 1×	PriDe 1.15×
`llama-2-7B`	−**77.55**	−76.09	+**12.44**	+12.29
`llama-2-chat-7B`	−**53.43**	−41.48	+2.52	+9.58
`llama-2-13B`	−58.63	−69.33	+**3.46**	+1.97
`llama-2-chat-13B`	−34.78	−42.66	+**2.97**	+2.68

Table 3: Percentage changes in RStd (bias) and accuracy before and after applying RBcorr (Ours) vs. PriDe. Higher percent decreases for RStd imply greater debiasing effect. Cases where our method results in larger improvement are highlighted.

We compare our method's efficacy to the PriDe method from (Zheng et al., 2024). Since their paper provides comprehensive debiasing results, including results from the Llama2 model family on the MMLU datasets, we run our correction method on the same models and datasets in order to directly use their provided results for comparison. Since their method scales efficacy with compute cost, while our method requires zero extra compute, we compare our method's results to the PriDe variation with the lowest cost, which carries a 15% (i.e. 1.15×) overhead compute. We also quantify our method's efficacy using the same bias metric (Zheng et al., 2024) use, i.e., RSTD or standard deviation over recalls.

Table 3 shows the accuracy and bias value changes achieved by performing `RBCorr` and PriDe on the same models and datasets. We observe that our method achieves comparable effects for both bias reduction and accuracy improvement while having zero overhead compute.

D Complete Per-Model Comparison of all Correction Methods

Table 4: Correction method effects (relative to baseline performance) on the instruction-only prompt.

Dataset	Model	CC Δ			BC Δ			**RBCorr Δ**		
		Acc.	TVD	RSD	Acc.	TVD	RSD	Acc.	TVD	RSD
ARITH (Yes/No)	F-3B	−13.17%	+0.1100	+0.2000	+3.00%	−0.1200	−0.1483	+2.58%	−0.1258	−0.1548
	F-3B-I	+0.42%	−0.0175	−0.0213	−0.33%	−0.0167	−0.0201	+0.00%	−0.0250	−0.0303
	F-10B	−1.00%	+0.0267	+0.0316	+1.67%	−0.0467	−0.0535	+1.25%	−0.0458	−0.0526
	F-10B-I	+0.17%	+0.0067	+0.0074	+0.33%	+0.0217	+0.0240	+0.42%	+0.0258	+0.0286
	L-8B	−1.33%	+0.0517	+0.0727	+4.25%	−0.1392	−0.1837	+4.17%	−0.1283	−0.1702
	L-8B-I	−9.83%	+0.1783	+0.2846	+3.33%	−0.1000	−0.1298	+3.42%	−0.1075	−0.1391
	L-70B	+7.58%	−0.0858	−0.1022	+5.83%	−0.0717	−0.0861	+5.92%	−0.0675	−0.0817
	L-70B-I	+0.25%	+0.0025	+0.0026	+0.33%	−0.0033	−0.0034	+0.00%	+0.0000	+0.0000
	G-27B	−9.00%	+0.1383	+0.1756	−1.58%	+0.0158	+0.0186	−0.17%	+0.0183	+0.0205
	G-27B-I	−0.33%	+0.0033	+0.0036	+0.00%	−0.0017	−0.0017	+0.08%	−0.0025	−0.0026
	G-12B	−12.50%	+0.1833	+0.2358	−1.00%	−0.0017	−0.0018	+0.00%	−0.0017	−0.0018
	G-12B-I	−0.33%	+0.0117	+0.0140	−0.75%	+0.0108	+0.0133	−0.25%	+0.0075	+0.0090
BABI (Yes/No)	F-3B	+5.50%	−0.0583	−0.1045	+13.50%	−0.2167	−0.3188	+13.92%	−0.2058	−0.3065
	F-3B-I	+2.00%	−0.0400	−0.0514	+1.50%	−0.0233	−0.0308	+1.25%	−0.0242	−0.0314
	F-10B	−0.25%	+0.0142	+0.0160	−0.08%	−0.0442	−0.0492	+0.67%	−0.0433	−0.0485
	F-10B-I	−0.67%	+0.0183	+0.0203	+0.83%	−0.0183	−0.0200	+0.83%	−0.0250	−0.0272
	L-8B	−14.92%	+0.1642	+0.2826	+3.58%	−0.1392	−0.1687	+3.25%	−0.1458	−0.1763
	L-8B-I	−11.67%	+0.1400	+0.2113	+1.58%	−0.1125	−0.1315	+1.83%	−0.1100	−0.1286
	L-70B	+1.67%	+0.0117	+0.0119	+1.50%	−0.0283	−0.0313	+1.92%	−0.0342	−0.0376
	L-70B-I	−0.25%	+0.0075	+0.0082	+0.17%	−0.0117	−0.0127	+0.50%	−0.0117	−0.0127
	G-27B	+9.00%	−0.1417	−0.1791	+8.50%	−0.1583	−0.1974	+8.58%	−0.1508	−0.1891
	G-27B-I	−0.08%	+0.0058	+0.0063	−0.08%	−0.0025	−0.0026	+0.17%	−0.0033	−0.0037
	G-12B	−5.17%	+0.1117	+0.1343	−0.50%	−0.0150	−0.0166	+0.00%	−0.0150	−0.0167
	G-12B-I	−0.25%	+0.0075	+0.0083	−0.33%	+0.0067	+0.0074	−0.08%	+0.0042	+0.0046
COMPS (Yes/No)	F-3B	+24.67%	−0.4183	−0.7702	+24.75%	−0.4258	−0.7796	+24.67%	−0.4183	−0.7702
	F-3B-I	−3.08%	+0.0658	+0.0890	+0.67%	−0.0350	−0.0441	+0.58%	−0.0392	−0.0492
	F-10B	+2.33%	−0.0383	−0.0517	+6.75%	−0.1275	−0.1610	+7.17%	−0.1250	−0.1582
	F-10B-I	−1.33%	+0.0267	+0.0351	+2.17%	−0.0767	−0.0945	+2.42%	−0.0775	−0.0956
	L-8B	−4.75%	+0.0875	+0.1433	+8.75%	−0.1875	−0.2529	+9.42%	−0.1792	−0.2431
	L-8B-I	−4.42%	+0.1042	+0.1318	+0.92%	−0.0392	−0.0461	+0.58%	−0.0342	−0.0402
	L-70B	+5.42%	−0.0692	−0.0911	+7.58%	−0.1458	−0.1791	+7.67%	−0.1417	−0.1745
	L-70B-I	+0.00%	−0.0200	−0.0225	+0.00%	−0.0250	−0.0281	−0.25%	−0.0225	−0.0253
	G-27B	+8.83%	−0.1783	−0.2298	+9.42%	−0.1792	−0.2308	+8.92%	−0.1808	−0.2327
	G-27B-I	+1.00%	−0.0117	−0.0148	+2.08%	−0.0342	−0.0417	+2.08%	−0.0258	−0.0322
	G-12B	+0.00%	+0.0167	+0.0200	+3.58%	−0.0692	−0.0847	+4.25%	−0.0775	−0.0946
	G-12B-I	+0.33%	−0.0067	−0.0077	+0.67%	−0.0167	−0.0191	+0.58%	−0.0158	−0.0182
EWOK (Yes/No)	F-3B	+2.08%	−0.0308	−0.0769	+6.75%	−0.3892	−0.6828	+6.33%	−0.4033	−0.7049
	F-3B-I	+1.25%	−0.0925	−0.1342	−0.33%	−0.0717	−0.1033	+1.08%	−0.0708	−0.1033
	F-10B	−1.08%	+0.0642	+0.0878	−0.08%	+0.0208	+0.0279	−0.42%	+0.0242	+0.0328
	F-10B-I	−0.33%	+0.0317	+0.0431	−0.17%	−0.0133	−0.0180	+0.17%	−0.0150	−0.0203
	L-8B	−8.00%	+0.1783	+0.3697	+2.42%	−0.1942	−0.3037	+1.83%	−0.2000	−0.3118
	L-8B-I	−14.58%	+0.3042	+0.5750	+1.83%	−0.0700	−0.0994	+1.83%	−0.0700	−0.0994
	L-70B	−4.75%	+0.0942	+0.1310	+0.42%	−0.0942	−0.1164	+1.50%	−0.0867	−0.1073
	L-70B-I	−0.42%	+0.0342	+0.0411	+0.17%	−0.0417	−0.0495	+0.33%	−0.0483	−0.0574
	G-27B	+11.17%	−0.2833	−0.4754	+10.25%	−0.3525	−0.5697	+10.58%	−0.3492	−0.5652
	G-27B-I	+0.17%	+0.0417	+0.0528	−0.50%	−0.0033	−0.0042	−0.25%	−0.0025	−0.0032
	G-12B	−10.08%	+0.3058	+0.4799	−0.58%	+0.0158	+0.0217	−0.17%	+0.0317	+0.0428
	G-12B-I	−0.67%	+0.0217	+0.0309	+1.00%	−0.0333	−0.0465	+1.33%	−0.0283	−0.0406
SNLI (0/1/2)	F-3B	+13.17%	−0.3342	−0.5697	+11.33%	−0.5250	−0.9365	+12.67%	−0.5583	−1.0037
	F-3B-I	+3.08%	+0.0308	+0.0523	+6.00%	−0.1683	−0.2098	+6.50%	−0.1683	−0.2062
	F-10B	+0.25%	+0.0158	+0.0779	+5.00%	−0.0583	−0.0749	+5.00%	−0.0558	−0.0804
	F-10B-I	+2.08%	−0.0367	−0.0344	+1.33%	−0.0258	−0.0130	+1.58%	−0.0242	−0.0163
	L-8B	+26.00%	−0.3475	−0.7066	+14.92%	−0.5158	−0.8219	+13.83%	−0.5125	−0.8157
	L-8B-I	+3.83%	+0.0050	+0.0964	+3.00%	−0.0475	−0.0049	+4.00%	−0.0400	−0.0156
	L-70B	+6.75%	−0.1200	−0.2940	+8.25%	−0.1217	−0.2774	+8.08%	−0.1333	−0.2921
	L-70B-I	+2.50%	−0.0383	−0.0568	+2.00%	−0.0367	−0.0444	+1.75%	−0.0308	−0.0403
	G-27B	+6.50%	−0.0750	−0.1669	+11.83%	−0.1633	−0.3238	+11.92%	−0.1658	−0.3298
	G-27B-I	+0.50%	−0.0008	−0.0044	+0.83%	−0.0025	−0.0079	+1.25%	−0.0058	−0.0129

Continued on next page

Dataset	Model	CC Δ			BC Δ			**RBCorr Δ**		
		Acc.	TVD	RSD	Acc.	TVD	RSD	Acc.	TVD	RSD
	G-12B	+5.25%	−0.0425	−0.0998	+9.83%	−0.1275	−0.2789	+10.33%	−0.1367	−0.2943
	G-12B-I	+2.92%	−0.0408	−0.0711	+2.42%	−0.0417	−0.0695	+2.50%	−0.0392	−0.0670
MNLI (0/1/2)	F-3B	+4.50%	−0.1825	−0.2561	+9.83%	−0.5933	−1.0169	+10.67%	−0.6150	−1.0499
	F-3B-I	+3.17%	+0.0350	+0.2941	+3.33%	−0.0500	+0.2076	+3.67%	−0.0483	+0.2034
	F-10B	−3.42%	+0.0408	+0.1281	+2.67%	−0.0758	−0.0976	+4.17%	−0.0733	−0.1093
	F-10B-I	+0.08%	−0.0000	+0.0553	+0.33%	−0.0317	+0.0126	+0.17%	−0.0325	+0.0137
	L-8B	+22.17%	−0.3283	−0.6718	+19.00%	−0.5750	−1.0008	+19.08%	−0.5667	−0.9765
	L-8B-I	+0.08%	+0.0550	+0.1316	+3.50%	−0.1108	−0.1727	+3.83%	−0.1092	−0.1695
	L-70B	+5.33%	−0.1533	−0.2964	+6.50%	−0.1842	−0.3526	+6.92%	−0.1683	−0.3325
	L-70B-I	−1.17%	+0.0008	+0.0001	+3.92%	−0.1058	−0.1793	+4.42%	−0.1075	−0.1845
	G-27B	+3.83%	−0.0517	−0.1241	+8.83%	−0.1933	−0.3687	+11.00%	−0.2092	−0.4032
	G-27B-I	+0.33%	−0.0067	+0.0056	+0.50%	−0.0217	−0.0342	+0.92%	−0.0233	−0.0386
	G-12B	+2.25%	−0.0508	−0.0807	+6.17%	−0.1625	−0.2847	+7.08%	−0.1658	−0.3019
	G-12B-I	−1.42%	+0.0033	+0.0535	−0.25%	−0.0050	+0.0037	+0.17%	−0.0067	−0.0025
HUM. (A/B/C/D)	F-3B	+0.92%	−0.0358	−0.0046	+1.83%	−0.1733	−0.1829	+2.58%	−0.1658	−0.1829
	F-3B-I	−0.42%	−0.0050	+0.0097	+0.33%	−0.0400	−0.0406	+0.83%	−0.0383	−0.0449
	F-10B	−0.58%	+0.0442	+0.0367	+0.42%	+0.0150	+0.0063	+0.33%	+0.0117	−0.0005
	F-10B-I	+0.42%	+0.0175	+0.1154	+3.67%	−0.0450	+0.0377	+3.33%	−0.0475	+0.0384
	L-8B	−3.75%	+0.1158	+0.0928	+1.17%	−0.0775	−0.1097	+1.83%	−0.0717	−0.1106
	L-8B-I	−6.25%	+0.1708	+0.2828	+0.33%	+0.0075	+0.0596	−0.17%	+0.0108	+0.0643
	L-70B	−5.50%	+0.0408	+0.0733	−0.25%	−0.0042	−0.0034	+0.67%	−0.0058	−0.0062
	L-70B-I	−2.50%	+0.0642	+0.1007	−0.33%	+0.0208	+0.0575	−0.83%	+0.0225	+0.0565
	G-27B	−4.17%	+0.1125	+0.1552	+0.58%	−0.0050	+0.0207	+0.67%	+0.0100	+0.0287
	G-27B-I	−0.92%	+0.0300	+0.0212	+0.83%	−0.0025	−0.0106	+0.75%	−0.0017	−0.0070
	G-12B	−6.33%	+0.0658	+0.1809	+0.50%	−0.0292	−0.0384	+0.58%	−0.0267	−0.0381
	G-12B-I	−0.58%	−0.0025	+0.0167	−0.33%	−0.0175	−0.0083	−0.25%	−0.0175	−0.0113
OTHERS (A/B/C/D)	F-3B	+2.92%	−0.0633	−0.0951	+3.33%	−0.2508	−0.3300	+3.08%	−0.2433	−0.3405
	F-3B-I	+0.08%	+0.0042	+0.0138	+1.42%	−0.0800	−0.1140	+1.00%	−0.0808	−0.1185
	F-10B	−0.92%	−0.0108	+0.0067	−0.67%	−0.0133	−0.0153	−1.17%	−0.0083	−0.0151
	F-10B-I	+2.17%	−0.0558	−0.0150	+4.08%	−0.1300	−0.1015	+3.42%	−0.1267	−0.0895
	L-8B	−0.25%	+0.0233	−0.0577	+3.58%	−0.0792	−0.2017	+3.33%	−0.0900	−0.2186
	L-8B-I	−2.92%	+0.1350	+0.2001	−0.50%	−0.0092	+0.0237	+0.08%	−0.0042	+0.0327
	L-70B	−4.58%	+0.1333	+0.1430	−0.75%	+0.0192	+0.0101	−0.83%	+0.0242	+0.0180
	L-70B-I	−1.92%	+0.0600	+0.0960	−0.67%	+0.0242	+0.0492	−0.42%	+0.0258	+0.0473
	G-27B	−3.92%	+0.1000	+0.1549	+0.25%	−0.0100	+0.0048	+0.17%	−0.0083	+0.0092
	G-27B-I	−2.08%	+0.0417	+0.0774	−0.08%	−0.0075	−0.0202	+0.08%	+0.0008	−0.0052
	G-12B	−3.58%	+0.0642	+0.1679	+2.58%	−0.0475	−0.0799	+3.17%	−0.0383	−0.0608
	G-12B-I	−0.33%	+0.0008	−0.0015	+0.08%	−0.0217	−0.0304	+0.17%	−0.0183	−0.0212
SOC. SCI. (A/B/C/D)	F-3B	+3.33%	−0.1025	−0.1355	+2.50%	−0.2308	−0.2815	+2.92%	−0.2433	−0.2953
	F-3B-I	+0.33%	+0.0017	+0.0120	+0.50%	−0.0225	−0.0293	+0.67%	−0.0208	−0.0261
	F-10B	+0.50%	−0.0008	−0.0157	+0.83%	−0.0083	−0.0358	+1.00%	−0.0133	−0.0278
	F-10B-I	+1.75%	−0.0292	+0.0245	+3.50%	−0.0875	−0.0309	+4.33%	−0.0867	−0.0304
	L-8B	−3.58%	+0.0850	+0.0634	+0.92%	−0.0500	−0.1250	+0.33%	−0.0408	−0.1054
	L-8B-I	−4.08%	+0.1608	+0.2053	+0.58%	+0.0100	+0.0251	+0.50%	+0.0108	+0.0209
	L-70B	−7.25%	+0.1325	+0.1337	−0.75%	+0.0367	+0.0271	−0.08%	+0.0250	+0.0214
	L-70B-I	−1.42%	+0.0708	+0.0988	−0.50%	+0.0292	+0.0539	−0.67%	+0.0300	+0.0552
	G-27B	−4.50%	+0.0767	+0.1206	−0.67%	+0.0108	+0.0199	−0.42%	+0.0133	+0.0224
	G-27B-I	−0.17%	+0.0308	+0.0543	+0.75%	+0.0083	+0.0053	+0.67%	+0.0083	+0.0058
	G-12B	−3.50%	+0.0717	+0.1418	+0.67%	−0.0208	−0.0395	+0.92%	−0.0225	−0.0417
	G-12B-I	−0.50%	+0.0008	+0.0118	+0.00%	−0.0117	−0.0083	−0.33%	−0.0133	−0.0123
STEM (A/B/C/D)	F-3B	+1.33%	−0.0417	−0.1838	+2.33%	−0.2167	−0.4105	+2.50%	−0.2192	−0.4191
	F-3B-I	+0.00%	−0.0200	−0.0204	−0.83%	−0.1000	−0.1875	−1.08%	−0.0850	−0.1654
	F-10B	−2.08%	+0.0083	−0.0011	−0.42%	−0.0200	−0.0478	−1.25%	−0.0250	−0.0483
	F-10B-I	−1.08%	−0.0083	−0.0073	+2.58%	−0.1083	−0.1547	+1.25%	−0.1242	−0.1654
	L-8B	−0.67%	+0.0325	−0.0591	+1.83%	−0.1800	−0.2978	+2.17%	−0.1808	−0.2834
	L-8B-I	−3.83%	+0.1808	+0.2460	−0.67%	−0.0383	−0.0531	+0.00%	−0.0283	−0.0389
	L-70B	−8.58%	+0.2300	+0.2482	−0.83%	+0.0292	+0.0320	−0.25%	+0.0333	+0.0428
	L-70B-I	−1.92%	+0.0908	+0.0731	+0.75%	+0.0008	+0.0076	−0.25%	−0.0025	−0.0068
	G-27B	−5.33%	+0.1442	+0.1812	−0.58%	−0.0267	−0.0243	−0.33%	−0.0317	−0.0362
	G-27B-I	−0.42%	+0.0158	+0.0596	+0.58%	−0.0108	−0.0090	+0.67%	−0.0108	−0.0113
	G-12B	−5.33%	+0.0692	+0.1407	−1.08%	−0.0525	−0.0722	−2.33%	−0.0483	−0.0715
	G-12B-I	+0.92%	−0.0125	−0.0083	+1.42%	−0.0450	−0.0565	+0.08%	−0.0433	−0.0619